Learning to Ignore: Fair and Task Independent Representations

Linda Helen Boedi and Dr. Helmut Grabner

ZHAW School of Engineering,
Zurich, Switzerland

Abstract. Training fair machine learning models, aiming for their interpretability and solving the problem of domain shift has gained a lot of interest in the last years. There is a vast amount of work addressing this topics, mostly in separation. In this work we show that they can be seen as a common framework of learning invariant representations. The representations should allow to predict the target while at the same time being invariant to sensitive attributes which split the dataset into subgroups. Our approach is based on the simple observation that it is impossible for any learning algorithm to differentiate samples if they have the same feature representation. This is formulated as an additional loss (regularizer) enforcing a common feature representation across subgroups. We apply it to learn fair models and interpret the influence of the sensitive attribute. Furthermore it can be used for domain adaptation, transferring knowledge and learning effectively from very few examples. In all applications it is essential not only to learn to predict the target, but also to learn what to *ignore*.

1 Introduction

In June 2020 MIT withdrew Tiny Images[1] a popular vision dataset as researchers found that it is socially biased. Biases in training data are a major issue for machine learning algorithms [20]. Especially, as they are increasingly used to make critical decisions. First, it is important to ensure that those systems are fair and do not discriminate certain groups. Secondly, interpretability of the decisions - "why" the system comes to that conclusion or how important a certain factor for decision making is - are desirable for better understanding. Thirdly, these trained models should generalize well. For many real world situations the data seen during training is different then the data which the models are applied to in production. Domain adaptation tries to transfer knowledge from a source domain (training set) to a particular target domain. In order to cope with these challenges many different approaches have been proposed over the last decade.

Fairness and domain adaptation seem to be very different topics, but the goal for both is actually learning invariant feature representations. In this paper we propose a yet simple approach for learning fair representation. A deep learning

[1] https://groups.csail.mit.edu/vision/TinyImages/, 2020/07/10.

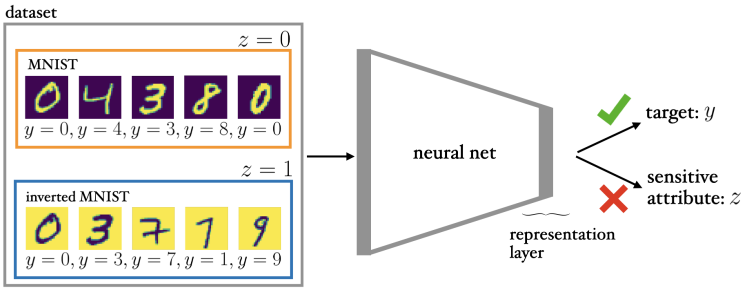


Fig. 1: Given a set of examples split into groups based on the sensitive attribute z, our learnt representation should allow to predict the target variable y, but at the same time not be able to predict the sensitive attribute z. In the example the digits should be predicted, however the origin (MNIST/inverted MNIST) is irrelevant and should be ignored.

model is forced to ignore certain information that would allow to draw conclusions about sensitive attributes (fair classifier) or certain areas (domain independent classifier). As seen in Fig. 1 the target variable should be still predictable, but at the same time it should not be distinguishable from which subgroup the examples were taken. The main insight is that similar feature representations for different groups or datasets do not allow us to differentiate between them anymore. To accomplish this, we introduce an affinity loss which is additionally used during the training of a model. Once a fair representation is established, the sensitive attribute can be added back and its impact measured. This paves the way for interpretability or causal reasoning. Furthermore, by reducing the distance between different domains in latent space a more general representation of the dataset is learned which helps to better generalize across domains.

Related Work. In order to make machine learning models ”fair”, works aim at modifying the feature representations of the data [30], the class label annotations [25] or the data itself [22]. Learning this latent representation includes an additional cross entropy classification loss [3], a decomposition loss [22], an additional hidden layer for adversarial optimization [1], distribution matching [21], using Variational Autoencoders [13], or by learning the representation as an adversarial minimax game [27]. The goal is not only to improve fairness but also to interpret how fairness is enforced. Such methods build on special network architectures [5, 12] or a combination of different machine learning algorithms [9]. For the domain adaptation task approaches make use of re-weighting the source samples to better match the target domain [11, 19], learning shared weights [28] or a common subspace [6], modifying the network architecture [15, 16], or using Generative Adversarial Networks [13, 7]. Interestingly, if causal aspects are taken into account, predictions can be improved [4, 24]. In the same line, recent work aims for analysing those areas in a common way [13, 23, 14].

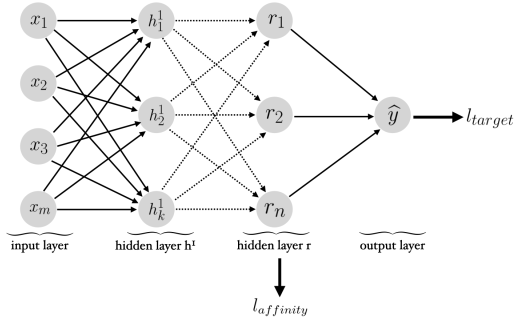


Fig. 2: Example architecture of a neural network with one ore more hidden layers h (indicated with the dotted lines). The last hidden layer $(r_1, ..., r_n)$, is reffed as representation and used for calculated our proposed affinity loss.

Contribution. Our main aim is not to beat any particular method for fairness or domain adaption, it is rather to highlight the commonalities. From a technical side, the most related work probably is the one by Ganin et al. [6] based on Zemel et al. [30]. However instead of adding a new gradient reversal layer, we simply reduce an additional affinity loss during training. Hence, our method can be easily applied to any existing network architecture including classification, regression tasks or auto-encoders[2]. Experiments demonstrate that a pretrained network with fixed weights can be simply debiased by adding a fair representation layer. While many approaches struggle with unbalanced datasets both in terms of the target and the sensitive attribute [3], our approach is not very negatively impacted. We are able to improve fairness, interpretability and domain adaptation within one very simple approach.

2 Learning Invariant Representation

Problem formulation. Let X be the entire data set. Each $x \in X$ is an example represented by m attributes and $y \in Y$ its corresponding target variable. Furthermore, let z be the sensitive attribute. We aim to learn a classifier $f_y : x \to y$ to predict the target variable from the attributes, but at the same time being *unable* to predict the sensitive attribute $f_z : x \to z$.

In order to achieve this we build a (low dim.) representation $g(x)$ which allows for predicting the target y but *not* the sensitive attribute z. Our aim is to make the representation as similar as possible with respect to the sensitive attribute. Hence, being invariant features. Many modern deep learning architectures can be seen as having such a representation layer, having the advantage that any pre-trained model can be used and later fine-tuned.

In the setting of training a neural network, typically a loss l_{target} (e.g., cross entropy) is minimized in order to predict the target. We propose to add another loss term $l_{affinity}$ which serves as regularizer. See a visualization in Fig. 2. The neural network is then trained on the combined loss

$$l_{total} = l_{target} + \lambda \cdot l_{affinity}. \tag{1}$$

[2] In this paper we focus on the classification tasks with one categorical sensitive variable.

If the weight λ of the affinity loss is set to zero, the model is trained normally without the new loss. If λ is very large the neural network optimizes on the affinity loss, ignoring the target loss. The fairness of a model increases with λ but might result in lower accuracy.

In the following we derive $l_{affinity}$. The sensitive attribute splits the dataset in one or more subgroups. For simplicity we focus on two subgroups in the following. Considering the two sets X_1, X_2 split by the sensitive attribute z. To be unable to distinguish between these two sets the following must hold

$$\forall x_1 \in X_1 \; \exists x_2 \in X_2 : g(x_1) = g(x_2). \tag{2}$$

In other words, for each sample there must be at least another sample with a different sensitive attribute having the same representation. Technically the loss is minimizing the closest distance of them.

The learnt representation should still allow to predict the target y. Trivial representations are avoided by the combination of the loss term (see above). However, our experiments show that it is beneficial to add a more strict constraint so that the two examples x_1 and x_2 are from the same class, i.e., $y_1 = y_2$. This avoids mixing up classes and yields significantly better performance. Averaging over all examples and all classes gives

$$l_{affinity} = \frac{1}{|Y||X_1|} \sum_{y \in Y} \sum_{x_1 \in (X_1 | y_1 = y)} \min_{x_2 \in (X_2 | y_2 = y)} d(g(x_1), g(x_2)), \tag{3}$$

where $d(\cdot, \cdot)$ is an arbitrary distance function.

Implementation Details. In order to implement Eq. (3) we use a nearest neighbor with the L_1 norm. To speed the training up we do not use the whole dataset, but only calculate the affinity loss on the mini-batches.

2.1 Experiments

We show the basic behavior of our method on an illustrative experiment based on the well known *MNIST* dataset of handwritten digits[3]. Additionally we created the *MNIST-I*, which contains all original *MNIST* images, but inverted. Together it forms our dataset where the sensitive attribute indicates if the digit originates from the *MNIST-I* or the original *MNIST*. As target we still want to predict which number is depicted.

We train a simple neural network with two 128- and a single 20-width ReLU hidden layer as representation layer. For training, a batch size of 128 samples is used and the weight of the proposed affinity loss is set to $\lambda = 0.01$.

Embedding. In order to analyze our learnt representation, we perform a t-Distributed Stochastic Neighbor Embedding (t-SNE) on the representation layer. It models the higher-dim. data by a low-dim. point such that similar objects lay closer together and dissimilar ones further away.

[3] http://yann.lecun.com/exdb/mnist/, 2020/07/10.

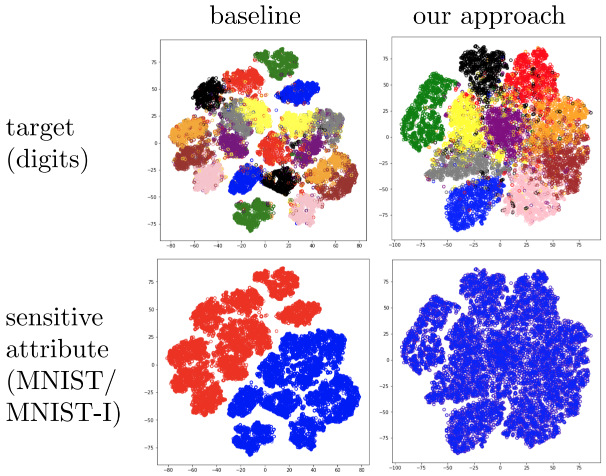


Fig. 3: 2-dim. t-SNE plot showing the learnt feature representations colored based on digit (top row) and on the sensitive attribute (bottom row). By using our approach, the representation shows distinguishable digit clusters but the dataset origin cannot be traced. Best viewed in colour!

Fig. 3 depicts the comparison of two models, trained without (baseline) and with the affinity loss. The baseline model learns two clusters for each digit (one normal and one inverted) and the groups can easily be separated by the sensitive attribute (MNIST/MNIST-I). In contrast, adding the proposed affinity loss into the training process shows that the two groups are highly overlapping, i.e., not being distinguishable anymore, creating a fair (more in Sec. 3) and more general (only one cluster per digit) feature representation. The digits can still be predicted very accurately as their clusters are kept very distinct from each other. The effect of generality is also used for domain adaptation in Sec. 4.

Predictions After training, we fix all hidden layers and do a normal retrain of the output layer. The output layer is not only trained to predict the digits but also if the sample comes from the original MNIST or inverted MNIST-I dataset. The model trained with our approach should struggle in learning the origin of the sample (inverted/not inverted). In fact, our fair model predicts the target class with 93% accuracy (4% higher than the baseline), whereas the sensitive attribution is hardly predictable anymore (around 57% accuracy). The baseline model can easily predict (nearly 100%) the sensitive attribute.

Hyperparameter. If the weight λ of the affinity loss is set to zero, we train the model only based on the target loss not focusing on making the model fair (see Tab. 1a, trained 5 times and averaged). For λ equals 0.01 we get the fairest model, as the numbers are predicted well (even better than the baseline model does) and the origin dataset of the input samples can hardly be predicted. For a λ of 0.1 the model gets as fair as possible by not learning anything at all. If the representation layer is chosen too small (smaller than 5 nodes) the model does not perform well in predicting the digits accurately (see Tab. 1b). If the number of nodes is getting too large (more than 50 in this example) the model gets unfair again suffering from the curse of dimensionality.

Table 1: Influence of Hyper-parameters on fairness and accuracy.

(a) Tuning λ balances accuracy vs. fairness.

λ	0	0.0001	0.001	0.01	0.1
accuracy digits	0.89 ± 0.02	0.9 ± 0.02	0.93 ± 0.01	0.93 ± 0.01	0.1 ± 0.005
accuracy sensitive	0.995 ± 0.005	0.99 ± 0.005	0.89 ± 0.02	0.57 ± 0.03	0.5 ± 0.005

(b) Too small representations (obviously) do not allow for good classification results. A too large embedding space suffers from the curse of dimensionality.

layer size	1	5	10	20	50	100
accuracy digits	0.1 ± 0.05	0.78 ± 0.03	0.93 ± 0.01	0.93 ± 0.01	0.95 ± 0.03	0.96 ± 0.04
accuracy sensitive	0.5 ± 0.05	0.54 ± 0.02	0.57 ± 0.02	0.57 ± 0.03	0.74 ± 0.04	0.85 ± 0.05

3 Fairness and Interpretability

Simply removing the sensitive attributes from a dataset is insufficient for eliminating their biases as there almost always exists an indirect influence of the sensitive information [17]. Our approach learns a feature representation of the data preserving general information but enforcing not to learn sensitive characteristic information.

After the fair training of the model we are able to interpret the classification and investigate in the influence of the sensitive attribute on the classification task [18]. We do so by reattaching the sensitive attribute z to the fair model again, see Fig. 4. For better interpretability the fair feature representation is is linearly combined, forming r. The reattachment of z and its interpretation is possible as r is trained to be independent of z (see also [2])

$$\hat{y} = f(w_r r + w_z z + b), \text{with } z \perp\!\!\!\perp r \tag{4}$$

where w_r and w_z are the learnt weights of the neural network, b the bias term and z the sensitive attribute and $f(\cdot)$ the transfer function (e.g., linear or sigmoid). The weights w_r and w_z of r and z, respectively, indicate how large the influence of the sensitive attribute on the classification is [18]. In the following experiments a model trained without the affinity loss using the same architecture as the fair one is reffed as baseline.

3.1 Fairness Measures

There are a lot of different fairness measures used for classification [26, 3, 8, 29]. Two commonly used ones are summarized in the following. Let $\hat{y}$ be the output of the classifier, y the true label and z the sensitive attribute.

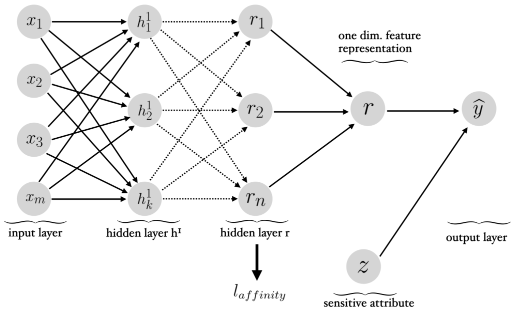


Fig. 4: A simple linear unit is added to the fair model representation. Furthermore the sensitive attribute z is reattached. As the both units are uncorrelated, we use the weights as interpretation for the importance of the sensitive attribute for the final classification.

Equality of Opportunity/ Equality Gap. The most common measure is the so-called equality of opportunity. It is reached if the groups z_1 and z_2 defined by the sensitive characteristic have equal true positive rates (TPR), i.e., $TPR_{z=z_1} = TPR_{z=z_1}$. The equality gap is then calculated as

$$P(\hat{y}=1|z=z_1,y=1) - P(\hat{y}=1|z=z_2,y=1) = |TPR_{z=z_1} - TPR_{z=z_2}|. \quad (5)$$

Parity Gap. The parity gap is calculated as independence between prediction $\hat{y}$ and sensitive attribute z for positive predictions, i.e.

$$|P(\hat{y}=1|Z=z_1) - P(\hat{y}=1|Z=z_2)| \quad (6)$$

For binary case of the sensitive attribute, in medical settings, it is the same as the average treatment effect (ATE) [2].

It is important to note that it is difficult to minimize all fairness metrics at the same time. The appropriate metric depends on the application, but most often the equality of opportunity is targeted. Be aware, that there are some trivial models which yield good results (very small gaps) such as models with very low TPR. For some tasks, the compromise may not even be possible, such as predicting whether someone can give birth. There is a clear causal relationship to the gender; thus, if this information (including implicit information) is removed, it becomes impossible for any classifier to make a correct prediction.

3.2 Experiment: Adult Dataset

The popular *Adult* income dataset[4] from the UCI is used for further experiments and the results are compared with other papers. The task is to predict whether or not an individual is earning more than \$50K per year. The samples are annotated with 14 different attributes from gender and educational level to number of work hours per week. The gender attribute is used as binary sensitive attribute for the affinity loss during training. The dataset is split into 26'049 samples for training, 6,512 samples for validating and 16,281 for testing.

[4] https://archive.ics.uci.edu/ml/datasets/adult, 2020/07/10.

Table 2: Our approach compared with a baseline and different other approaches concerning accuracy and fairness on the Adult dataset.

approach	**accuracy**	**parity gap**	**equ. gap**	**equ. gap (TNR)**
baseline	0.85	0.18	0.088	0.072
ours	0.82	0.065	0.02	0.015
Quadrianto'18 [21]	0.81	-	0.04	-
Adel'19 [1]	0.89	0.13	-	-
Zemel'13 [30]	0.82	-	-	-
Quadrianto'17 [22]	0.84	-	0.017	-
Beutel'17 [3]	0.82	0.12	0.07	0.04
Louizos'17 [13]	0.82	-	0.05	-
Xie'17 [27]	0.84	-	-	-

We train our model with a batch size of 512 samples using a network with one hidden layer of 128 and another one with 20 nodes on whose feature representation the affinity loss is calculated. We compare our model with several state-of-the-art approaches as well as against the baseline. Results are summarized in Tab. 2. Our approach achieves a similar fairness level compared to other approaches. Consistently, our feature representation promoted fairness criteria with only a small penalty in accuracy even though the dataset is heavily skewed.

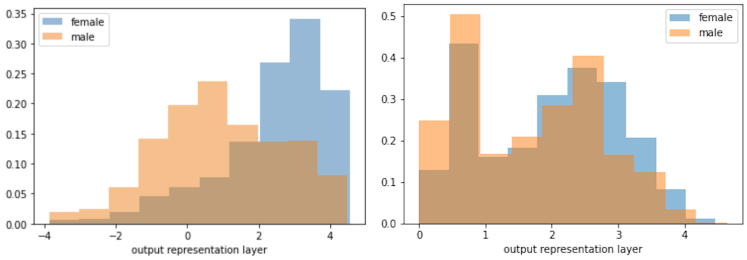
Fair Representation. We train two baseline models once inputting all the features including the sensitive attribute and once removing this attribute. We assume that simply removing the sensitive attribute does not help to omit the gender bias [17]. The performances are compared with our fair model and if the sensitive attribute is added back, see Fig. 4. For each model we retrain the last layer (hidden layers are fixed) to predict once the gender of the input samples and once the income. Results are summarized in Tab. 3. The accuracies of the baseline models lay very close together, which shows that information about the gender attribute is indeed still hidden in the input data. Our model is not able to classify the genders correctly just labeling almost all samples as male. The performance of our model with reattached gender attribute is similar to the baseline models. This shows that the sensitive attribute helps the model to perform better.

The histograms of the fair one-dim. representation (see Fig. 4) for the male and female samples in Fig. 5 show a very similar distribution. Hence, this supports the assumption that our model contains a fair representation of the data.

Interpretablity of the sensitive attribute. The influence of the gender attribute on the classification of the input samples concerning the income is evaluated. The weight of the fair one-dim. feature representation r and the weight of the input of the sensitive attribute z are compared (see Fig. 4). We consider the samples where the income is larger than $50K. The weight for z lays around 4, whereas the weight for r is approximately 1, thus much smaller. This shows that the input of the sensitive attribute has indeed a large influence on the classification as it holds highly valuable information.

Table 3: Accuracy and fairness measures predicting income and gender attribute (z) on the Adult dataset with different training approaches.

	baseline		**our approach**	
	with z	without z		z reattached
accuracy income	0.85	0.85	0.81	0.83
parity gap	0.16	0.17	0.05	0.15
equality gap	0.069	0.074	0.0075	0.1
accuracy female	0.46	0.25	0.027	1.0
accuracy male	0.84	0.86	0.98	1.0



(a) baseline

(b) our approach

Fig. 5: Histogram of the outputs of r (see Fig. 4) for male and female samples. Without enforcing any fair training the learnt distributions carry a lot of information about the sensitive attribute (a) in contrast to our approach with overlapping distributions (b), not allowing to trace back the sensitive attribute.

3.3 Experiment: CelebA

The *CelebA* image dataset[5] is significantly more complex than the *MNIST* or *Adult* dataset. This record contains a total of 202,599 images of celebrities, each with 40 attributes. 162,770 images are used for training, 19,867 for validating and the rest for testing. The annotated attributes reflect appearance of the celebrities as well as the emotional state (e.g. smiling), gender, attractiveness and age. The gender attribute is used as a binary sensitive characteristic and attractiveness as a target label for the classification of the images.

As model we use a fixed VGG19 net trained on imagenet (to speed up the training process and reduce complexity) and an additional hidden layer with 124 nodes. Tab. 4 compares results with different weights λ and shows that we can in fact debias the pretrained VGG net. The *CelebA* dataset is heavily skewed; around $\sim 77\%$ of the images showing women are labeled as attractive, compared to $\sim 23\%$ of men. If λ is strong enough, the influence of the skew on the fairness disappears. The downside is the decrease of accuracy to only $\sim 55\%$ as the TNRs for female and male are getting low. Please note, the comparison with Quadrianto [22] is not too accurate as our baseline already has a lower accuracy.

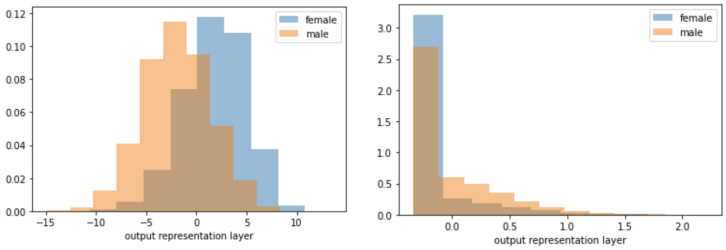
[5] http://mmlab .ie.cuhk.edu.hk/projects/CelebA.html, 2020/07/10.

Table 4: Fairness of different models on the CelebA dataset. A lower weight of λ keeps the accuracy higher, but improves fairness only to a limited amount. A higher weight decreases the accuracy significantly but makes the model fair.

	baseline	**our approach**			Quadrianto'18[22]	
		$\lambda = 0.05$	$\lambda = 0.07$	$\lambda = 0.1$	**fair**	**baseline**
accuracy	0.73	0.64	0.62	0.55	0.8	0.8
parity gap	0.47	0.23	0.22	0.0038	-	-
equality gap	0.31	0.25	0.23	0.018	0.19	0.34

Table 5: Accuracies and fairness measures predicting gender attribute z and income on the CelebA dataset with different training approaches.

	baseline		**our approach**	
	with z	without z		z reattached
accuracy	0.73	0.74	0.62	0.7
parity gap	0.47	0.49	0.22	0.63
equality gap	0.31	0.32	0.23	0.59
accuracy female	0.86	0.84	0.88	1.0
accuracy male	0.83	0.85	0.25	1.0



(a) baseline

(b) our approach

Fig. 6: Histogram of the outputs of r for male and female samples for *CelebA*.

Fair representation & Intepretability. The histograms of the the fair one-dim. representation (see Fig. 4) for the male and female samples in Fig. 6 show a very similar distribution, supporting the assumption that our model contains a fair representation of the data. The influence of the gender attribute on the classification, see Tab. 5, is checked with the same approach as described in Sec. 3.2. The similar accuracies of the baseline models show that information about the gender attribute is indeed still hidden in the input data. The reattached sensitive attribute helps the fair model to perform better in classifying faces as attractive. This can also be seen in the weight of the sensitive attribute z with around 1.5, compared to the one of the fair one-dim. feature representation r with around 0.7.

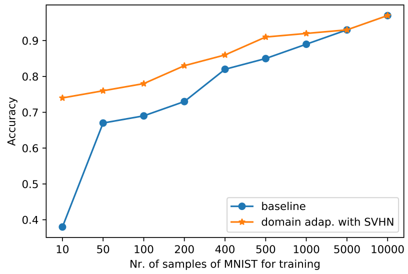


Fig. 7: Accuracy of MNIST with different number of training samples from the target domain (MNIST-R). Our approach (orange) clearly outperforms the simple baseline, especially when only few data from the target domain is provided.

4 Domain Adaptation

Data used for training a model might not be the same as during test time. This is a big problem for robust real world applications. The sensitive attribute relates now to the different domains or environments [4]. As seen in Sec. 2.1 we enforce to learn a more general feature representation and to ignore domain specific attributes. This is leveraged to learn representations which are generic across (related) domains and hence would generalize better.

From MNIST to MNIST-R. Additionally to the *MNIST* dataset we created the *MNIST-R* dataset containing all original MNIST images rotated by 30 degrees. We train a simple neural net with two 128- and one 20-width ReLU representations. MNIST is used as source while the performance is measured on MNIST-R (target). Inspired by Heinze-Deml et al. [10] few samples of the target set are used to improve the performance. We compare the results in Tab. 6 with a baseline model trained on the same amount of samples (20 per class) of the target dataset using data augmentation. Our approach can keep up with data augmentation, respectively even performs better if the imbalance in the amount of samples used during training becomes larger. It can better leverage the structure in the source data and map it to the target domain than simple data augmentation which relies on predefined transformations.

From SVHN to MNIST. The Street-View House Number (SVHN) dataset[6], contains house numbers from Google Street View. The challenge of the SVHN dataset is the structured clutter in the background of images. A Convolutional Neural Network (CNN) with two double-Convolutional layers containing 32 and 64 nodes, respectively, is used. A 20-dim. feature representation on top of this architecture is applied to calculate the affinity loss. Results and comparison to Ganin et al. [6] are shown in Tab. 7. The affinity loss does indeed improve the performance on the target dataset with only a little amount of samples. The performance of our model trained on the SVHN dataset with 10 MNIST samples reaches an accuracy of around 75% and can be compared with Ganin et al. [6]. In comparison, the baseline model achieves only an accuracy of 3% (see Fig. 7) on

[6] http://ufldl.stanford.edu/housenumbers/, 2020/07/10.

Table 6: Domain adaptation on MNIST (source) and MNIST-R (target). Displayed are the accuracies on the MNIST-R dataset after training (training dataset indicated in the header).

# source samples	**only source dataset**	**+ 200 target samples and data aug.**	**our approach (+ 200 target samples)**
1,000	0.52	0.76	0.78
10,000	0.67	0.76	0.82

Table 7: Performance of models trained with the affinity loss on SVHN (source) and MNIST (target). For training a small amount of labeled target samples is used. Ganin et al. [6] uses only unlabeled target samples for training.

		accuracy SVHN (source domain)	**accuracy MNIST (target domain)**
	only SVHN	0.91	0.11
baseline	+ 100 MNIST samples	0.91	0.11
	+ 200 MNIST samples	0.91	0.13
our approach	+ 10 MNIST samples	0.91	0.75
	+ 100 MNIST samples	0.91	0.80
	+ 200 MNIST samples	0.92	0.85
	Ganin et al. [6]	-	0.74

the same data. If there are only a few MNIST samples available, the neural net trained with SVHN and our affinity loss outperforms the baseline model trained solely on the same amount of MNIST samples.

5 Discussion and Conclusions

We proposed a new approach for learning invariant feature representation. The main idea is to bring the feature representation of different distributions closer together by introducing an additional loss. We applied this strategy to three different areas: fairness, interpretability and domain adaptation. Our proposed method can be used for different model architectures as well as for readjusting the feature representation of existing, already trained models. Experiments show that the equality gap can be significantly reduced while the accuracy is still kept at an acceptable level. The results are comparable with state-of-the-art methods for each task. We demonstrate how to understand how a sensitive attribute influences the classification of an input sample. A challenge in our approach is to efficiently find the nearest neighbors in the embedding space. We rely on effective, approximated methods here. Not much thematized in this paper is that our approach allows using multiple source and target datasets. Thus a model can be trained to be fair regarding multiple attributes. A further extension might be using real-valued attributes as sensitive attributes.

References

1. Adel, T., Valera, I., Ghahramani, Z., Weller, A.: One-network adversarial fairness. In: AAAI Conference on Artificial Intelligence. AAAI (2019)
2. van Amsterdam, W.A.C., Verhoeff, J.J.C., de Jong, P.A., Leiner, T., Eijkemans, M.J.C.: Eliminating biasing signals in lung cancer images for prognosis predictions with deep learning. npj Digital Medicine 2 (2019)
3. Beutel, A., Chen, J., Zhao, Z., Chi, E.H.: Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations. Tech. rep., Google Research (2017)
4. Bühlmann, P.: Invariance, Causality and Robustness. Tech. rep., ETH Zurich (2018)
5. Chattopadhyay, A., Manupriya, P., Sarkar, A., Balasubramanian, V.N.: Neural Network Attributions: A Causal Perspective. In: International Conference on Machine Learning (2019)
6. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research 17 (2017)
7. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D.: Domain Generalization for Object Recognition with Multi-task Autoencoders. In: International Conference on Computer Vision (2015)
8. Hardt, M., Price, E.P., Srebro, N.: Equality of opportunity in supervised learning. In: Conference on Neural Information Processing Systems (2016)
9. Hartford, J., Lewis, G., Leyton-Brown, K., Taddy, M.: Counterfactual Prediction with Deep Instrumental Variables Networks. Tech. rep., Microsoft Research and University of British Columbia (2016)
10. Heinze-Deml, C., Meinshausen, N.: Conditional Variance Penalties and Domain Shift Robustness. Tech. rep., ETH Zurich (2017)
11. Huang, J., Smola, A., Gretton, A., Borgwardt, K., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: Conference on Neural Information Processing Systems (2006)
12. Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., Welling, M.: Causal effect inference with deep latent-variable models. In: Conference on Neural Information Processing Systems (2017)
13. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.: The Variational Fair Autoencoder. In: International Conference on Learning Representations (2015)
14. Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., Mooij, J.M.: Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions. In: Advances in Neural Information Processing Systems (2017)
15. Mancini, M., Porzi, L., Bulò, S.R., Caputo, B., Ricci, E.: Boosting Domain Adaptation by Discovering Latent Domains. In: Conference on Computer Vision and Pattern Recognition (2018)
16. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified Deep Supervised Domain Adaptation and Generalization. In: International Conference on Computer Vision (2017)
17. Pedreschi, D., Ruggieri, S., Turini, F.: Discrimination-aware Data Mining. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2008)

18. Peters, J., Janzing, D., Schölkopf, B.: Elements of Causal Inference: Foundations and Learning Algorithms. MIT Press, Cambridge, MA (2017)
19. Pinheiro, P.O.: Unsupervised Domain Adaptation with Similarity Learning. In: Conference on Computer Vision and Pattern Recognition (2018)
20. Prabhu, V., Birhane, A.: Large image datasets: A pyrrhic win for computer vision? Tech. rep., UnifyID Inc. (2020)
21. Quadrianto, N., Sharmanska, V.: Recycling privileged learning and distribution matching for fairness. In: Conference on Neural Information Processing Systems (2017)
22. Quadrianto, N., Sharmanska, V., Thomas, O.: Discovering Fair Representations in the Data Domain. In: Conference on Computer Vision and Pattern Recognition (2018)
23. Schumann, C., Wang, X., Beutel, A., Chen, J., Qian, H., Chi, E.H.: Transfer of Machine Learning Fairness across Domains. Tech. rep., Google (2019)
24. Singh, H., Singh, R., Mhasawade, V., Chunara, R.: Fair Predictors under Distribution Shift. In: Conference on Neural Information Processing Systems (2019)
25. Thanh Luong, B., Ruggieri, S., Turini, F.: k-NN as an Implementation of Situation Testing for Discrimination Discovery and Prevention. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
26. Verma, S., Rubin, J.: Fairness Definitions Explained. In: IEEE/ACM International Workshop on Software Fairness (2018)
27. Xie, Q., Dai, Z., Du, Y., Hovy, E., Neubig, G.: Controllable Invariance through Adversarial Feature Learning. In: Conference on Neural Information Processing Systems (2017)
28. Yang, J., Yang, R., Hauptmann, A.G.: Adapting svm classifiers to data with shifted distributions. In: International Conference on Data Mining Workshops (2007)
29. Zafar, M.B., Valera, I., Rodriguez, M.G., Gummadi, K.P.: Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In: International World Wide Web Conference (2017)
30. Zemel, R., Ledell, Y.., Wu, ., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: International Conference on Machine Learning. vol. 3. ICML (2013)